Multipath Routing in Path Computation Element (PCE): Protocol Extensions and Implementation

Xiaomin Chen, Yuesheng Zhong, Admela Jukan
Technische Universität Carolo-Wilhelmina zu Braunschweig
Email: chen@ida.ing.tu-bs.de, y.zhong@tu-bs.de, jukan@ida.ing.tu-bs.de

Abstract—**The Path Computation Element (PCE) architecture as well as the PCE communication protocol (PCEP) have been standardized by the IETF. Extensions have also been proposed to enrich PCE architectures for various scenarios, including inter-domain and multi-layer networks. As of today, all research and standardization in PCE concepts have focused on single path computation, while concurrent computation of multiple paths, commonly referred to as multipath routing, has not been addressed yet. This paper presents the first PCE implementation and PCEP protocol extensions with multipath routing capability. We implement the multipath extensions in an open source PCE emulator and study the performance of the system in carrier grade networks which can be optical or Carrier Ethernet networks. We show that PCE with multipath routing capability performs well in different network conditions in terms of network load and network size. We also show that multipath extensions in PCEs can cause larger signaling delay comparing with single path computation. However, it maintains in the order of milliseconds, even in large networks, which is applicable in practice.**

I. Introduction

Path Computation Element (PCE) has been proposed to facilitate on-demand path computation for dynamic connection requests with Quality of Service (QoS) requirements. The PCE systems work in a server-client fashion, i.e., the PCE server computes an optimal path upon the request from a Path Computation Client (PCC) based on the information maintained in its Traffic Engineering Database (TED). The communication between PCE servers and clients is based on the PCE communication protocol (PCEP) [1]. The scalability and capability of constrained path computation makes PCE especially attractive to carriers. With the latest extensions on related protocols, PCE has become the de-facto standard in provisioning dynamic connection-oriented services, especially in the circuit switching networks with strict QoS requirements. Whereas the PCE frameworks proposed to date have exclusively considered single path computation, the increasing demand for flexible services with high bandwidth requirement is driving the evolution of PCE towards multipath techniques.

Extending PCE with multipath computation capability is particularly interesting in the area of Carrier Ethernet and optical networks. Today, most of the carrier-grade systems are migrating to 100Gbps Ethernet (100GbE) technologies. While 100GbE serial interfaces are still in lab-trial phase, parallel transmission over multiple interfaces over low speed channels is a valid solution [2]. It has been proposed to use multiple MAC layouts ranging from 10x10Gbps, 27.5x4Gbps, 3x40Gbps and so on, utilizing low speed optical channels [3]. With capability of multipath routing, PCE is a viable candidate to support this migration process. Another motivating example is in the area of high performance computing, where applications require connection services across transport networks with extremely high bandwidth requirements. For instance, streaming of uncompressed single view 4K video requires a connection with bandwidth of 15.2Gbps. It is difficult to support such applications with a single path, especially with high network load. Parallel transmission on multiple paths appears as a natural solution. More generally, multipath routing is known for its benefits in load balancing and fault tolerance, which also drives PCE frameworks evolve towards multipath computation capability.

In this paper, we propose multipath routing extensions in PCE, including extensions in PCEP protocol and signaling process. We prototype the multipath PCE system by implementing the proposed extensions on an open-source, vendor-independent PCE emulator [4]. The presented prototype allows for any carrier-specific routing mechanisms. To illustrate the performance of the multipath PCE systems, we implemented a multipath routing mechanism in this paper which includes *multipath computation* and *bandwidth allocation*. We show that some multipath specific constraints, such as maximal acceptable differential delay, need to be carefully considered when implementing multipath extensions in the PCE standards. Also, the PCE communication protocol, PCEP, needs to be extended to allow for requesting multipath computation and returning of multiple Explicit Route Objects (EROs). Our study is the first step towards these standardized extensions for circuit switching networks, especially optical networks and Carrier Ethernet. We show that the presented multipath PCE achieves a good performance on various network topologies and loads, which implies that enabling multipath routing in PCEs is feasible and widely deployable.

The rest of the paper is organized as follows: Section II presents the PCE background and the related work. The deployment example and protocol extensions are presented in Section III. Section IV presents the performance evaluation and Section V concludes the paper.

II. Background and Related Work

A. *PCE architecture in a nutshell*

The Path Computation Element (PCE) is positioned as a server which can perform constrained path computation based on the topology information stored in its Traffic Engineering

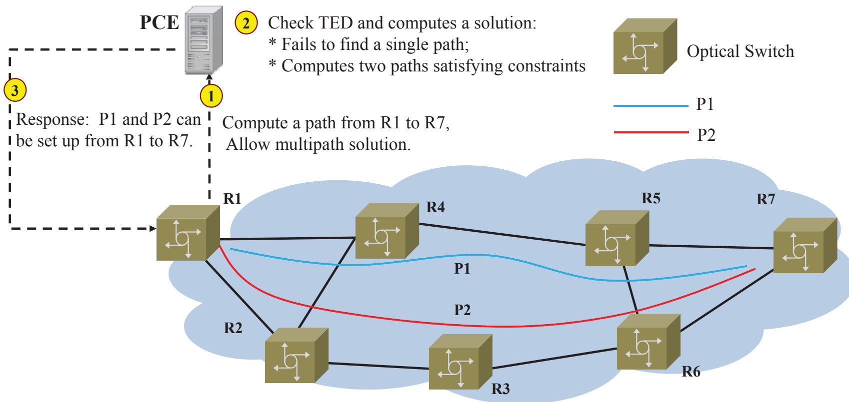

Fig. 1. Multipath PCE deployment in an example transport network

Database (TED). The TED of a PCE contains information of the governed network, including QoS parameters such as available capacity, link delay, etc. and is typically updated via network control plane (e.g. GMPLS [5]) by including the ERO into the Resource Reservation Protocol (RSVP) Path message [6]. The use of a TED allows the PCE to compute a best path that fulfills all QoS requirements. It is especially useful in provisioning services in transport networks (such as optical networks) where strict QoS requirements are specified. Path computation in a PCE architecture is initiated by a Path Computation Client (PCC) by sending a Path Computation Element Communication Protocol (PCEP) [1]. Further extensions allow a PCE to be a client to another PCE in order to extend the reach of the optimal path computation to the multi-layer [7] and multi-domain [8] scenarios. For example, in an inter-domain scenario, the PCE in an upstream domain acts as a client to a downstream domain PCE, requesting an optimal path to the destination and this process is repeated along a domain chain to compute the inter-domain path [8].

B. PCE Communication Protocol (PCEP)

The communication protocol used between two PCE peers is Path Computation Element communication Protocol (PCEP). Seven unique message types are defined in the PCEP, which are specified as follows. Open and Close messages are used to initialize and close the connection between clients and PCE server; Path Computation Request message (PCReq) is used to send a request, to which the response is sent back in Path Computation Response message (PCRep); and finally Keepalive, Notification and Error are used in order to convey additional information to remote peers. Extensions to the PCE architecture almost always require updates to the basic PCE protocol specification defined in [1]. While the initial set of protocols are targeted to compute a single path. Efforts have also been paid to extend the PCE architecture and related specifications to address issues such as policy integration [9], monitoring [10] and point-tomultipoint path computation [11] as well as extensions to support Wavelength Switched Optical Networks (WSON) [12].

C. Multipath Routing in PCE

Despite that a significant body of PCE extensions have been proposed, integration multipath routing in PCE has rarely been studied, especially in the context of protocol and implementation. Primary work to connect PCE and multipath routing was reported in [13] which proposed to use PCE to facilitate inter-domain multipath routing in circuit switching networks. In [13], the inter-domain multipath computation relies on the network abstraction technologies to compose multiple virtual routing planes. Standard PCE is deployed to compute a single path on each routing planes, instead of tackling multipath extension in PCEs. To the best of our knowledge, this paper is the first attempt to propose multipath extensions in PCE which covers both protocol extensions and implementation.

D. Differential Delay Issue in Multipath Routing

Differential delay issue in multipath routing has to be considered in extending PCE with capability of multipath routing. When multiple paths are used to support a connection request, the delay of each path may be different, leading to the dis-order of traffic in the destination node. Solutions have been proposed to counter the differential delay issue in multipath routing. For instance, ITU-T G.709 [14] has suggested that the realignment process has to be able to compensate a differential delay of at least $\pm 125us$. A commercial framer device can support $250us$ differential delay using internal memory and up to $128ms$ using off-chip Synchronous Dynamic Random Access Memory (SDRAM)[15], [16]. In this paper, we consider the maximal acceptable differential delay as a new constraint in the PCEP protocol, in align with the solutions in commercial devices and standardization efforts.

III. Implementation Overview

The implementation overview of multipath routing extensions in PCE is presented in Figure 1, where an optical network is used as the example network. To better understanding multipath routing in PCE, an example is illustrated, where a connection is required to be established from R1 to R7. A typical multipath computation workflow is summarized as follows: The Path Computation Client (PCC), i.e., R1 here,

```
<PCReq Message> :: = <Common Header>
                     < Multipath>
                     <End-points>
                     <Metric-list>
Where <Metric-list> ::= <Maximal acceptable differential delay>
                        <End-to-end delay>
                        <Bandwidth>
```

Fig. 2. PCEP request message with multipath extension

```
<PCRep Message> :: = <Common Header>
                     <End-points>
                     <Path-list>
                     <Metric-list>
Where  <Path-list>  :: =  [<ERO><Bandwidth>]
Where <Metric-list> ::= <Maximal acceptable differential delay>
                        <End-to-end delay>
```

Fig. 3. PCEP reply message with multipath extension

sends a path computation request with QoS constraints such as bandwidth and delay to the PCE which governs the network (Step1). Upon receiving the request, PCE checks the available resources in its TED and computes a path. In case that the PCE fails to compute a single path with given constraints, the PCE sends back a response to inform PCC (R1) that there is no path available (Step2). In the conventional PCE architectures, the connection request would be blocked, while multipath extensions in PCE allows the PCC to request a multipath solution (Step3). In the multipath computation request, the maximal acceptable differential delay (DD) is included as the special constraint of multipath routing. Upon receiving the multipath request, the PCE calculates a multipath solution based on the set of constraints given by PCC. In this example, two paths are computed, namely P1 and P2. Afterwards, the explicit information of two paths are sent back to R1.

A. Multipath Extension in PCEP Protocol

In this section, we proceed to describe the extensions required in the PCEP protocol to enable multipath routing. As described above, seven messages have been defined in IETF standard [1]. To integrate multipath routing in PCE, two messages need to be modified, namely, Path Computation Request message (PCReq) and Path Computation Response message (PCRep). The multipath extensions to PCEP protocol are shown in Figure 2 and Figure 3. When PCC requests a multipath connection, it sends out an extended PCReq message with the multipath specific constraint, i.e., *Maximal acceptable differential delay*, as shown in Figure 2. Upon receiving the multipath computation request, PCE initiates computation of multiple paths based on the resource information in its TED. If a set of paths can be found that fulfill the QoS requirements, including bandwidth, end-to-end delay and maximal differential delay, PCE returns a PCRep message with *Path-list*, specifying a set of Explicit Route Objects (EROs) and bandwidth assigned on each path, as shown in Figure 3.

B. PCE Signaling for Multipath Computation

The multipath computation signaling is shown in Figure 4. The signaling for the communication between the PCE client and PCE server follows the specification standardized in RFC5440 [1]. As the first step, the PCC (here R1) initiates a PCEP session with the PCE, which involves the exchange of Open and Keepalive messages. The client then sends a Path Computation Request (PCReq) which specifies source and destination nodes (endpoints) and QoS constraints (bandwidth, delay etc) to PCE. The PCE attempts to compute a single path and returns the description of the path as an Explicit Route Object within the Path Computation Response Message (PCRep). Here the PCE fails to compute a single path, it returns a response message implying *No Path*. The PCC then sends a multipath computation request message in the Path Computation Request (PCReq) with multipath extension, i.e., maximal acceptable differential delay (DD) in Figure 4. If the PCE succeeds to compute a set of paths based on the embedded multipath routing algorithm, satisfying all the given QoS constraints, it returns a PCRep message that contains a set of EROs and allocated bandwidth on each path. In the example shown in Figure 1, the Path-list contains two paths, i.e., P1 and P2. Therefore, the EROs included in the PCRep message are R1-R4-R5-R7 and R1-R2-R3-R6-R7.

C. Multipath Routing Algorithm

In the multipath PCE implementation, path computation is based on the routing algorithms implemented in the *Computation Module* of the open source PCE emulator [4]. It is flexible to integrate any carrier-specific routing algorithms in the implementation. Here, we propose a mechanism which is composed of two heuristic algorithms in the presented multipath PCE, namely, *multipath computation* and *multipath bandwidth allocation*. The multipath computation algorithm is used to compute a set of paths between the given endpoints (source and destination pair), while the bandwidth allocation algorithm is used to select paths with QoS requirements and allocate bandwidth on each path.

Before going into the details of the multipath routing algorithms, we first define the notations that are used in this section. Network topology is represented as $G(V, E)$ where V and E are the set of network nodes and links, respectively. Service endpoints, i.e., source and destination nodes are represented as $< s, d >$. QoS constraints are denoted as B (Bandwidth), D (End-to-end delay) and DD (Maximal acceptable differential delay). Paths computed from the multipath computation algorithm are sorted in the descending order of bandwidth in a path set, denoted as $\mathcal{P} = \{P_1, P_2, P_3, ...\}$. Bandwidth allocated on path P_i is denoted as t_i. Delay of path P_i is denoted as D_i. The multipath routing algorithms implemented in the multipath PCE are presented as follows.

1) Multipath Computation Algorithm: As shown in Alg.1, the multipath computation starts from computing all path segments from source (s) to its neighboring nodes. All the path segments initiated from source node are sorted in the descending order of bandwidth. In each iteration, all path

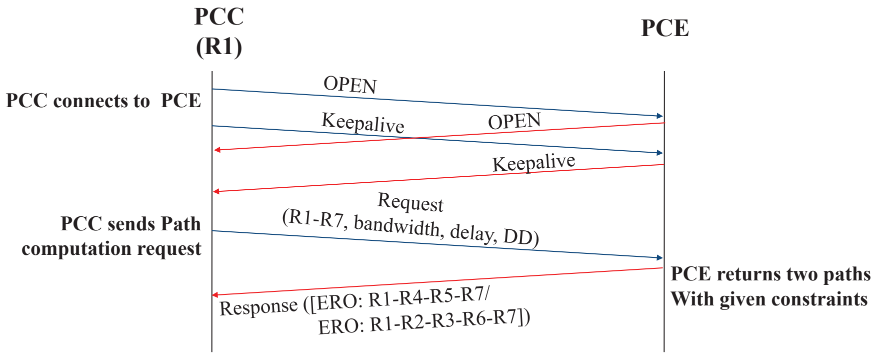

Fig. 4. PCE signaling for multipath routing; Note that messages such as Close and Error are not shown in this signaling flow

segments are extended to the neighboring nodes. All the current path segments in $\mathcal{P}$ are also sorted in the descending order of bandwidth. In the case that the path segment selected is terminated at the destination, the next path segment in line is chosen. For each possible extension for the chosen path, we check if a loop is formed in the path segment. This is done by checking if the new vertex that the path segment is extended to exists in the path segment. The new path segment is then inserted in $\mathcal{P}$ if it is valid.

2) Multipath Bandwidth Allocation Algorithm: We implement a heuristic algorithm for multipath bandwidth allocation, as shown in Alg.2. It starts from taking the path with maximum bandwidth in $\mathcal{P}$ and assumes it as the path with highest delay, denoted as $\tilde{P}$. All paths with delay less than $\tilde{P}$ in $\mathcal{P}$ are selected and sorted in the descending order of delay in a temporary path set $\mathcal{P}'$. It goes on checking the bandwidth and differential delay constraints. If all the constraints are satisfied, it outputs the paths and bandwidth that will be allocated on each path, i.e., t_i and P_i.

IV. Performance Evaluation

In this section, we present the performance evaluation of the presented multipath PCE system. In a network under study, we randomly select a node as a Path Computation Client (PCC) and each client generates path computation requests which are sent to the PCE server. Each client has the same average inter-arrival time (1 sec) and we vary the total number of clients active in order to vary request rate on the PCE server. The performance measures are based on the traffic load in the network (Erlang) defined as $u{*}h$, where u is connection arrival rate and h is the mean connection holding time. Blocking ratio used is defined as the percentage of the blocked connections in the total number of incoming connection demands.

All the network topologies studied in this paper are assumed to have a link capacity of 40Gbps. The bandwidth required by the connection requests is randomly generated, varying between 3Gbps and 6Gbps. For each network load, average around 10000 connections are generated in order to obtain a statistically relevant value. In all the results shown in the following, the maximum acceptable differential delay is set to be 128ms, which is in align with the commercial available devices [15], [16]. The performance of multipath PCE system is evaluated with three parameters, namely: network load, the number of paths and topology size.

A. Effect of Network Load

In this study, our goal is to show the performance of multipath PCE at different network load condition. The performance of PCE with single path computation is also studied under the same condition to provide a benchmark. Atlanta network topology [17] with 15 nodes and 22 links are used in the study; and all the possible paths are computed for each path computation request, i.e., N in Alg.1 is unlimited. In addition, Shortest-Path-First algorithm is also used in PCE to compute a single path as a performance benchmark.

As it can be seen in Figure 5, the PCE implemented with multipath extensions can successfully set up connections even with high network loads. When network load is 50 Erlang, only 638 connections are blocked out of 10000 connection requests, which results in a blocking ratio of 6.38%. However, PCE with single path routing results in a dramatic performance decrease with increasing network load. A blocking ratio of 22.36% is observed at 50 Erlang. It can observed from this study that PCE with multipath extensions has a rather stable performance in serving path computation requests with increasing network load. Under the same network condition, the conventional PCE with single path may fail to compute a path, especially when network load is high.

We further show the overhead caused by multipath extensions by comparing signaling delay of PCE with multipath extensions and PCE with single path computation. The signaling delay is defined as the time from PCC sends a path computation request till it receives a response from the PCE server. At each network load, an average signaling delay is obtained from around 10000 connection requests. As shown in Table I, the multipath extensions in PCE increase the signaling delay, comparing with PCE with single path. It takes about 10 ms in average for a multipath PCE to compute a connection for a path computation request while single path PCE takes about 2 ms in average to finish a path computation. However, the signaling delay in the PCEs with multipath extensions is still in the order of milliseconds, while significantly increasing

Algorithm 1: Multipath path computation algorithm

```
Input: G(V, E), < s, d, D >
count = 0; P = ∅
For all v_i ∈ V s.t. there exists a link e_j ∈ E from s to
v_i, create a path from s to v_i and insert in P
Sort P in decreasing order of bandwidth
while count < N do
    count = 0 for (i = 0 to P.length) do
        if (P[i].destination == d) then
            Increase count by 1
            path = P[i]
            remove P[i] from P
            foreach (e_j ∈ E attached to path.destination)
            do
                Let the vertices at the end of e_j be v_k
                and path.endVertex
                if (v_k ∉ path.vertices) then
                    Create a new path by extending path
                    with e_j
                    Insert new path in P[i]
                    Sort P in decreasing order of
                    bandwidth
    if (no more paths can be extended) then
        break;
    for (i = 1, .., P.length) do
        if (D_i > D) then
            Remove path P_i from P
Output: P
```

Algorithm 2: Multipath Bandwidth Allocation

```
Input: P, < s, d, B, DD >
for i = 1, .., P.length do
    P̃ = P_i
    for j = i, .., 1 do
        if D_j ≤ D_i then
            Put in a new path set P'
    Sort P' in the descending order of delay
    for i = 1, .., P'.length do
        bandwidth = bandwidth + B_i;
        if bandwidth ≥ B then
            for k = 1, .., i do
                if (d_P̃ − d_k) ≤ DD then
                    Output calculated paths and exit
Output: {t_i, P_i ∈ P}
```

the acceptance of connections requests.

B. Number of Paths

While computing all possible paths between a given source and destination pair facilitates to obtain an optimal solution, it consumes a lot of processing power and leads to a large signaling delay. We thereafter limit the number of paths computed for a path computation request, i.e., limit the value of N in Alg.1. As shown in Figure 6, the number of blocked connection requests is increased when N is limited to be 6. It is especially true when network load is high. However, the increase of blocking ratio is rather slight. For instance, around 1% increase is observed when network load is 50 Erlang. It is due to the fact that a connection solution computed by the PCE servers for a path computation request is generally less than five paths in this study. Therefore, limiting the number of paths in the multipath routing algorithms does not significantly affect the network performance regarding blocking ratio. On the other hand, it can decrease the overhead caused by the multipath extensions, i.e., reducing the signaling delay of multipath PCE. As shown in Tab.I, the signaling delay is about 6 ms with $N = 6$, instead of 10 ms with unlimited N.

TABLE I
SIGNALING DELAY OF PCE WITH SINGLE PATH AND PCE WITH MULTIPATH EXTENSIONS (ATLANTA)

Load (Erlang)	Multipath, N unlimited (Avg. ms)	Multipath, $N = 6$ (Avg. ms)	Single path (Avg. ms)
30	10.920	6.328	2.369
35	9.914	6.237	2.353
40	9.122	6.306	2.316
45	11.835	6.654	2.322
50	11.982	6.759	2.386

C. Effect of Network Size

Finally, we study the effect of network size where the PCE with multipath extension is applied. We use the Germany50 network topology, 50 nodes and 88 links [17] and study the performance of the PCE with multipath extensions. The network settings are the same as Atlanta network. We observe that the network performance improvement, in terms of blocking ratio, in Germany50 follows the same trend as it is in Atalanta (Sec.IV-A). Therefore, we will not repeat the discussion here, due to the space limit. However, the network size can significantly affect the signaling delay of the PCEs. PCEs with and without multipath extensions have larger signaling delays in Germany50 network than in the Atlanta network. Tab.II quantitatively shows the average signaling delay of multipath PCE in the Germany50 network. It can be seen that multipath PCE needs longer time for signaling in a large network. For instance, the signaling delay of PCE with multipath extensions is 68.299 ms at 35 Erlang when N is unlimited, while only 9.914 ms is required for signaling in the Atlanta network at the same network load. However, it can be reduced to 37.757 ms by limiting the number of paths ($N = 6$). In the Germany50 network, PCE with single path computation also experiences longer signaling delay, comparing with signaling delay in the Atlanta network. About 5.8 ms is required for signaling in the Germany 50 network, while the signaling delay is about 2.3 ms in the Atlanta network. However, the signaling delay of multipath PCE in a large network is still within in the order of milliseconds, even at high network loads.

V. CONCLUSION AND DISCUSSION

In this paper, we proposed multipath extensions in PCE and presented the first implementation of multipath PCE based on in an open-source PCE emulator. We proposed the

TABLE II
SIGNALING DELAY OF PCE WITH SINGLE PATH AND PCE WITH MULTIPATH EXTENSIONS (GERMANY50)

Load (Erlang)	Multipath, N unlimited (Avg. ms)	Multipath, $N = 6$ (Avg. ms)	Single path (Avg. ms)
30	66.254	35.354	5.819
35	68.299	37.757	5.829
40	64.638	33.084	5.846
45	65.248	32.208	5.835
50	68.034	35.798	5.825

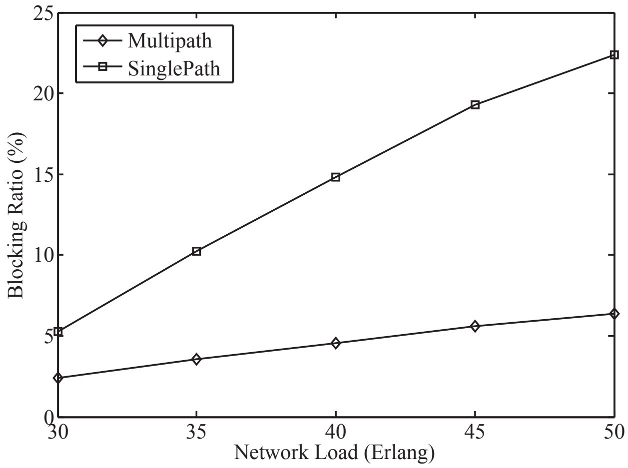

Fig. 5. Blocking ratio of PCE with single path and PCE with multipath extensions vs. different network loads (Atlanta)

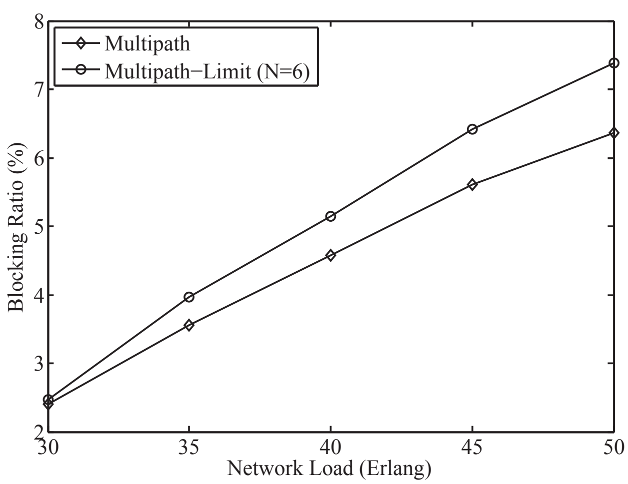

Fig. 6. Blocking ratio of PCE with multipath extensions with unlimited N and with $N = 6$ at different network loads (Atlanta)

protocol extensions and studied the performance of the PCEs with multipath routing capability in different network loads and network size. We showed the overhead, i.e., signaling delay, caused by the multipath extensions in the PCEs is acceptable, while the blocking ratio of connection requests can be significantly reduced. We also suggested that the number of paths can be limited in the multipath routing algorithms to reduce the overhead caused by multipath extensions with a slightly scarification in terms of blocking ratio. Finally, we also showed that the presented multipath PCE system is scalable in large networks, with average signaling delay in the order of milliseconds. Therefore, it is widely deployable and it can be beneficial for transport networks with emerging new applications, such as optical and Carrier Ethernet networks.

VI. ACKNOWLEDGMENT

This work has been partially supported by GEYSERS (FP7-ICT-248657) project funded by the European Commission through the 7th ICT Framework Program.

REFERENCES

[1] J. Vasseur and J. L. Roux, “Path computation element communication protocol,” RFC 5440, Mar. 2009.

[2] A. Gumaste, “Towards a transponder for serial 100gigabit ethernet using a novel optical serdes,” 2008.

[3] *Carrier Sense Multiple Access with Collision Detection(CSMA/CD) Access method and Physical Layer Specifications*, IEEE Std. 802.3ba, 2010.

[4] Open source path computation element emulator. [Online]. Available: http://pce.ida-cns-group.net/

[5] L. Berger and D.Fedyk, “Generalized mpls (gmpls) data channel switching capable (dcsc) and channel set label extensions,” RFC 6002, Oct. 2010.

[6] R.Braden, L.Zhang, S. Berson, S. Herzong, and S. Jamin, “Resource reservation protocol (rsvp)– version 1 functional specification,” RFC 2205, Sept. 1997.

[7] E. Oki, T. Takeda, J. L. Roux, and A. Farrel, “Framework for pce-based inter-layer mpls and gmpls traffic engineering,” RFC 5623, Sept. 2009.

[8] J. Vasseur, R. Zhang, N. Bitar, and J. L. Roux, “A backward-recursive pce-based computation (brpc) procedure to compute shortest constrained inter-domain traffic engineering label switched paths,” RFC 5441, Apr. 2009.

[9] I. Bryskin, D. Papadimitriou, L. Berger, and J. Ash, “Policy-enabled path computation framework,” RFC 5394, Dec. 2008.

[10] J. Vasseur, J. L. Roux, and Y. Ikejiri, “A set of monitoring tools for path computation element (pce)-based architecture,” RFC 5886, Jun. 2010.

[11] Q. Zhao, D. King, F.Verhaeghe, T. Takeda, Z. Ali, and J. Meuric, “Extensions to the path computation element communication protocol (pcep) for point-to-multipoint traffic engineering label switched paths,” RFC 6006, Sept. 2010.

[12] Y. Lee, G. Bernstein, J. Martensson, T. Takeda, and T.Otani, “Pcep requirements for wson routing and wavelength assignment,” IETF Internet Draft, Jul. 2011.

[13] X. Chen, M. Chamania, and A. Jukan, “A backward-compatible inter-domain multipath routing framework,” in *Proc. IEEE INFOCOM Computer Communications Workshops*, april 2011, pp. 133 –138.

[14] “Interfaces for the optical transport network (OTN),” ITU-T Recommendations. [Online]. Available: http://www.itu.int/rec/T-REC-G.709/e

[15] S. Huang, S. Rai, and B. Mukherjee, “Survivable differential delay aware multi-service over sonet/sdh networks with virtual concatenation,” in *OFC/NFOEC*, Mar. 2007, pp. 1–3.

[16] “Intel ixf19301 datasheet.” [Online]. Available: http://www.intel.com/design/network/products/optical/tsp/IXF19301.htm

[17] Osurvivable fixed telecommunication network design. [Online]. Available: http://sndlib.zib.de/home.action